Metaheuristic Traceability Attack against SLMAP, an RFID Lightweight Authentication Protocol

Julio C. Hernandez-Castro*, Juan E. Tapiador‡, Pedro Peris-Lopez†, John A. Clark‡ and El-Ghazali Talbi§
* School of Computing, Portsmouth University
Julio.Hernandez-Castro@port.ac.uk
† Information and Communication Theory Group, Delft University of Technology
P.PerisLopez@tudelft.nl
‡Department of Computer Science, University of York
jet@cs.york.ac.uk, jac@cs.york.ac.uk
§INRIA Futurs, Villeneuve d'Ascq, Lille
El-ghazali.Talbi@lifl.fr

***Abstract*—We present a metaheuristic-based attack against the traceability of an ultra-lightweight authentication protocol for RFID environments called SLMAP, and analyse its implications. The main interest of our approach is that it is a complete black-box technique that doesn't make any assumptions on the components of the underlying protocol and can thus be easily generalised to analyse many other proposals.**

I. Introduction

The idea of attacking cryptographic protocols by means of metaheuristic procedures is relatively new. Two very relevant works in this area are that of Knudsen *et al.* [1] and that of Clark *et al.* [2], where the authors were the first to use various heuristic techniques for protocol cryptanalysis. Even though these results had quite an impact on the security of Identification Protocols, they both were more focused on solving the underlying NP-hard problem (the Permuted Perceptron Problem, PPP [7]) than in addressing the protocol itself.

There is also a more recent but quite preliminary work [5] of some interest but limited relevance because its authors were only able of cryptanalysing a toy protocol (a scaled-down and simplified version of SASI [6]). Conversely, the related area of evolving or automatically designing cryptographic protocols by means of different heuristic techniques has seen considerable success too, and some notable (and even human competitive) results have been already attained [3], [4].

The rest of the paper is organised as follows. In the next section, we present a general model for the metaheuristic attack on cryptographic protocols. After this, in Section III we describe a recently proposed authentication protocol for RFID environments called SLMAP, which could be attacked as described in Section IV. Finally, in Section V we draw some conclusions and propose possible improvements together with future research lines.

II. General Attack Model

The main idea behind our approach is to *transform the cryptanalysis of a security protocol into a search problem*, where a large number of different search metaheuristics can be applied. In general, during this search we will try to find which are the secret state values (keys, nonces, etc.) of some subset of the parties involved in the protocol.

This, of course, could be done in various ways, but the most natural approach (while seriously limited[1]) is to measure the cost of the tentative set of secret values by the *proximity* [2] of the messages produced by these tentative solutions to the real public messages generated and exchanged during the actual protocol execution.

Most cryptographic protocols should exchange one or more messages to accomplish their intended objective(s) (authentication, key exchange, key agreement, etc.), and in the vast majority of cases these messages are sent via an insecure or public channel that can be easily snooped.

In our attack model, the cryptanalyst will generally try to infer the secret values that the two parties intend to hide by exploiting the knowledge of the exchanged messages. In a robust, secure, well-designed cryptographic protocol, even states that are very close to the real state should not produce messages that are very close (for any useful distance definition) of the real public messages. This should be done, typically, by means of a careful design and message construction based on the use of some highly-nonlinear cryptographic primitives such as block ciphers or hash functions.

Unfortunately, new proposals in the field of lightweight cryptography, which are intended towards very computationally constrained environments (such as low-cost RFID systems) cannot use classical cryptographic primitives such as hash functions [12]. That implies that many of the blossoming ultra-lightweight authentication protocols that are currently being proposed could be potentially open to attacks based in metaheuristic search that would be much harder to mount had these proposals been armed with classical cryptographic algorithms. In addition to this, many of these new protocol

[1] Protocols could be explicitly designed to make this task impossible or much harder by being many-to-one, in the sense of allowing a large number of possible secret states to exchange exactly the same public messages during a protocol execution, but this seems not to be the case in most recent ultra-lightweight authentication proposals.

[2] This will usually be measured by the cost function guiding the metaheuristic search.

proposals are supposed to offer new security services that are far from the classical ones, including some that are not yet well understood and even lack an unanimously accepted definition by the security community (i.e. traceability [11]). That only makes things harder for protocol designers, and more unlikely for their proposals to be secure enough to be deployed.

III. Description of the SLMAP protocol

In 2007, Li and Wang proposed a very interesting ultra-lightweight mutual authentication protocol intended for very low-cost RFID tags [8]. This protocol was called SLMAP and only used very efficient operations, such as bitwise XOR and addition mod 2^{96}. It avoided using costly operations such as multiplication, hash computation or exponentiation (as there is the common assumption that they cannot be carried out over low-cost RFID tags) and the generation of random numbers (nonces) was performed by the RFID reader (generally assumed not to have severe computational limitations).

The SLMAP protocol is briefly described in the following, where R represents a reader, T represents a tag, IDS^m stands for an index pseudonym in session m, ID is tag's private ID, K_i^m represent tag's secret keys during session m, and r is a nonce. All variables have a 96-bit length.

1) $R \rightarrow T : hello$

2) $T \rightarrow R : IDS^m$

3) With IDS^m, the reader finds in the backend database the tag's secret values ID, K_1^m, and K_2^m.

4) R generates nonce r to construct messages A and B as follows

   $$A = IDS^m \oplus K_1^m + r$$
   $$B = IDS^m + K_2^m \oplus r$$

   where $\oplus$ stands for the usual bitwise addition modulo 2, $+$ represents addition modulo 2^{96}, and the $\oplus$ operation is assumed to have a higher precedence than modular addition. The reader sends to the tag the concatenation of A and B

   $R \rightarrow T : A \| B$

5) From A and B the tag can obtain the value of r and verify that the reader knows K_1^m and K_2^m. Then it locally computes C and sends its value.

   $T \rightarrow R : C$ with
   $$C = (IDS^m + ID \oplus r) \oplus (K_1^m + r) \oplus (K_2^m + r)$$

   where ID is the tag secret ID, only known to him and authorized readers, a constant value typically set at the manufacturing stage which cannot be transmitted in clear because this will allow for trivial traceability attacks. It is thus the value IDS, or index pseudonym, which changes in a seemingly random fashion after each authentication session to difficult tracking attacks.

6) R verifies C and, if it is equal to the result of its local computation, updates IDS^m, K_1^m and K_2^m, and computes and sends D:

   $$IDS^{m+1} = (IDS^m + K_1^m \oplus r + (ID + K_2^m)) \oplus r$$
   $$K_1^{m+1} = K_1^m \oplus r + (IDS^{m+1} + K_2^m + ID)$$
   $$K_2^{m+1} = K_2^m \oplus r + (IDS^{m+1} + K_1^m + ID)$$

   $R \rightarrow T : D$ with

   $$D = IDS^{m+1} \oplus (ID + r) + (K_1^m \oplus K_2^m \oplus r)$$

To the best of our knowledge, SLMAP has received no attacks yet. As is usually the case with ultra-lightweight authentication protocols designed for very resource constrained environments, it is intended to be at least secure against passive attacks. Much more powerful active attacks would probably be possible, although these are generally precluded in the definition of most of proposals [14], [15], [16]. We focus, then, on passive attacks because these directly question the security objectives of the protocol, and simultaneously make the weakest assumptions about the attacker capabilities. Passive attacks are also clearly least risky for the attacker, since interference may otherwise be detected.

IV. Cryptanalysis of the SLMAP protocol

After eavesdropping one single session, any attacker will have access to the values IDS, A, B, C, D exchanged during this session. As the description of the SLMAP algorithm is public, we can start from a random set of secret values $\{K_1', K_2', r', ID'\}$ and run the protocol over them to see what messages A', B', C', D' do they generate.

Then, using a metaheuristic technique, we can search for those that minimise the distance between the candidate and the real exchanged messages. In this case, we will use a Simulated Annealing technique, which is a metaheuristic technique that is extremely efficient and has some ability to avoid becoming quickly trapped in local minima.

As an additional justification for the use of this approach, we can mention that it was the one employed in all three relevant works in the area [1], [2], [5] published to date.

A. Cost Function

Different definitions of the cost function have been tried, and the most successful was, by far:

$$f_S = \sum_{i=0}^{i=N} wt(M_i \oplus A_i) \cdot 96^{3-i} \tag{1}$$

where M_i stands for the real (snooped) message and A_i is its approximation as computed from the values of candidate state S.

For our particular problem, equation (1) has the following form:

$$\begin{aligned} f_S &= \sum_{i=0}^{i=3} wt(M_i \oplus A_i) \cdot 96^i \\ &= wt(M_3 \oplus A_3) + wt(M_2 \oplus A_2) \cdot 96 \\ &+ wt(M_1 \oplus A_1) \cdot 96^2 + wt(M_0 \oplus A_0) \cdot 96^3 \end{aligned}$$

Where $wt(\cdot)$ stands for the Hamming weight.

It is important to note here that, according to the definition presented in [2], the use of this straightforward cost function will not correspond to the application of what authors name a *warping* technique. The correct set of secret values will always lead to a global minimum.

This is a quite simple cost function that reflects the intuitive idea of establishing a kind of lexicographical order between the distances of the generated messages A', B', C' and D' to those observed in the actual protocol run. The general strategy here is to try to force that the first generated message A', should be really very close to the real one A, and in fact in many cases we obtained a Hamming distance of zero between the two. Then, we should look for very good approximations to B, and only when stalled in this process we will start to refine the value of C'. With the parameter setting shown in Table I, in most of the cases the Simulated Annealing algorithm did not had time to properly work on minimising the distance between D' and D.

For this reason, we believe that it is very likely that higher values for the number of moves at a given temperature, together with a larger cooling rate and initial temperature will, at the cost of some efficiency, lead to better results. Probably, less direct approaches will also work well, especially the use of *warping* techniques. In this particular case, however, the most simple formulation worked sufficiently well so we did not were tempted to introduce any unnecessary complexity.

B. Predicting the IDS

The approach for forecasting the value of IDS is, then, to use a Simulated Annealing heuristic for trying to minimise the cost function and only then use the best seen values of the secret state $\{K_1', K_2', r', ID'\}$ to run a SLMAP protocol over them and see what value they *predict* for the IDS in the next authentication session.

After extensive experimentation, a set of parameters was found to be a fair compromise between efficiency (the SA algorithm will be employed many times) and efficacy. These are given in Table I.

One important characteristic of our attack is that it is successful after eavesdropping only one authentication session, which is a very economic requirement compared with those of other passive attacks.

We have performed multiple simulations for measuring the effectiveness of this approach. In all the cases, we initialised all secret and public values of the protocol to random values generated with the Mersenne Twister [13] pseudorandom number generator.

TABLE I
SA PARAMETERS FOR TRACING SLMAP

Initial Temperature	10
Cooling Rate	0.9
Max. Failed Cycles	∞
Moves at Temperature	500
Final Temperature	2^{-5}

The last element needed to run the Simulated Annealing algorithm over the SLMAP protocol is a neighbourhood definition. The source code in Python of our neighbourhood implementation is shown in the following:

```
def neighbour(I):
    L=[I[0], I[1], I[2], I[3], I[4]]
    for i in range(1,3):
        index=randint(1,4)
        pos=randint(0,96)
        L[index]=L[index]^(1<<pos)
    return L
```

We have run 20 experiments, each one consisting in 50 Simulated Annealing executions to try minimise the cost function described in Equation (1).

If a forecast algorithm is not working on this particular problem, and because of the simplest probabilistic arguments, we can expect around $50 \cdot P\{Binomial(96, 0.5) \leq 48\} \simeq 0.5406 \cdot 50 \simeq 27.03$ of these SA runs to find a *good* (i.e. at a Hamming distance of $\frac{96}{2} = 48$) or below and, around 22.97 of them finding *bad* ones (i.e. at a Hamming distance strictly higher than 48).

Any statistically significant departure from this behaviour means the metaheuristic search is being successful, and this is exactly what we have observed in our experiments. The number of *good* approximations was consistently (through all the 20 experiments) and significantly (average of 40.3 instead of 27) better that expected.

When in each of the 20 experiments we computed the majority vector of the 50 approximations, this vector was always quite a good approximation of the real IDS value, having an average of 57.25 correct bits over the 48 correct bits that one should expect from a random approximation. Some of these results are shown in Table II.

C. Traceability Attack

1) The Model: The *good* approximations to the next IDS tag value obtained in the described way can be exploited to mount a traceability attack, following the untraceability definition as proposed by Juels and Weis [9], and later used by Phan in his attack against SASI [10].

This untraceability model is briefly described in the following, where we will restrict ourselves to passive attacks:

- Adversary ***A*** interacts with a set of tags ***T*** and readers ***R***

TABLE II
ATTACK RESULTS FOR 20 RUNS, AFTER OBSERVING 1 AUTH. SESSION

Experiment	Good Approx.	Bad Approx.	Correct bits	Correlation	Rand. Corr.	Exp. Result
1	46	4	60	0.23591	-0.00532	Success
2	37	13	60	0.24760	-0.08020	Success
3	36	14	53	0.10418	0.02094	Success
4	39	11	56	0.13681	-0.01610	Success
5	44	6	59	0.22518	-0.03496	Success
6	34	16	53	0.10919	-0.12903	Success
7	41	9	56	0.17157	0.12330	Success
8	43	7	62	0.29247	0.08456	Success
9	45	5	60	0.24967	-0.06447	Success
10	42	8	58	0.19593	0.19375	Success
11	44	6	58	0.21009	0.0994	Success
12	35	15	53	0.10707	-0.11736	Success
13	38	12	53	0.09923	-0.02172	Success
14	47	3	61	0.26238	-0.04895	Success
15	35	15	53	0.10629	0.14811	Fail
16	47	3	66	0.36751	-0.16724	Success
17	44	6	59	0.21093	0.03115	Success
18	39	11	55	0.14885	0.08340	Success
19	31	19	52	0.10013	-0.01543	Success
20	39	11	58	0.20798	0.03845	Success
Averages	**40.3**	**9.7**	**57.25**	**0.19**	**0.01**	**95% Success**

- He can perform the *Execute(R,T,i)* query that allows him, by means of eavesdropping, to get access to a honest execution of the protocol session *i* between reader ***R*** and tag ***T***
- He can also perform a *Test(i,*T_0*,*T_1*)* query to mount an untraceability test. After executing this query, depending of a random bit $b \in \{0,1\}$ the attacker ***A*** is given $IDS_b \in \{IDS_0, IDS_1\}$ corresponding to tags $\{T_0, T_1\}$. The attacker succeeds if he can guess the random bit b with a probability better than flipping an unbiased coin.
- This probability determines $Avd_A^{UNT}(k)$, where k is a security parameter generally depending on the length of the secret state, and could be expressed as $Avd_A^{UNT}(k) = |Pr[$***A*** guesses b correctly$] - \frac{1}{2}|$
- We say that an RFID protocol achieves untraceability *(UNT)* if

  $$Avd_A^{UNT}(k) < \varepsilon(k)$$

  for some negligible function $\varepsilon(x)$

2) Results: It is clear that under this model (and under any other reasonable one, for that matter) the SLMAP protocol does not achieve untraceability, which is one of the main aims of RFID protocols to avoid tracking attacks.

For seeing this, it suffices to observe that once the attacker ***A*** has performed an *Execute(R,T,i)* query and, as a result of this, has eavesdropped the values of messages A, B, C and D he can launch the metaheuristic attack as proposed in Section IV to obtain many *good* approximations to the next IDS value.

After that, by performing a *Test(i,*T_0*,*T_1*)* query he will be able to compute the correlation[3] between his approximations and the two values $\{IDS_0, IDS_1\}$, knowing that higher values will likely correspond to that of the eavesdropped tag.

We have carried out this attack to approximate the value of $Avd_A^{UNT}(k)$, and heuristically (over 20 experiments) obtained a success probability of around 95%, which implies a non negligible attacker advantage of around 0.95-0.5=0.45. The average correlation between the IDS of the eavesdropped tag and the majority vector after a metaheuristic run was of 0.19, while that between the approximation and another random IDS was, as expected, very close to zero (0.01). These results are depicted in Table II.

It can be seen that, although no approximation is perfect, all of them are much closer to the real IDS value than what would have been expected at random. Combining all these approximations into a system of equations for obtaining even a closer value to the IDS is, although not technically very challenging, left for future works.

Each of the 50 run experiments takes approximately 30 minutes in a very modest portable computer. They are completely parallelizable.

3) Discussion: It is remarkable that all the correlations measured between the real IDS value and that obtained after launching the metaheuristic algorithm (a simulated annealing in this case) are positive, indicating that there is actually a successful learning process.

On the other hand, roughly one half between the corre-

[3] We refer here to the Pearson or *centered* correlation between two binary vectors, measured as their cosine once they have been centered to have a zero average, i.e.

$$corr(A,B) = \cos\widehat{AB} = \frac{x \cdot y}{\|x\| \cdot \|y\|} \qquad (2)$$

lations computed between the IDS and random values are, as expected, negative. This apparently makes easy for this attack scheme to produce and attacker advantage greater to zero, but things are not so simple. Firstly, always producing approximations with a positive correlation value is far from trivial. Most *natural* and *intuitive* cost functions fail to achieve even this which, one should note, is neither a sufficient nor a necessary condition for a successful attack.

Furthermore, for achieving a success ratio significantly higher that a 75%, it is necessary not only to steadily construct positive correlated outputs, but also to get them more correlated with the IDS value than expected from positive random correlated values (i.e. $|\rho_0| > \rho_1$ around 75% of the times).

For SLMAP, it is not so difficult to find other cost functions that lead to a non-negligible attacker advantage, but in very few cases this advantage is larger than $\frac{1}{4}$ (corresponding to the 75% ratio).

4) Traceability Pseudocode: The general traceability attack algorithm is described in Figure 2.

```
0. Snoop a SLMAP run between Tag $T_0$ and R, get $IDS^n$, A, B, C, D
1. For $i = 0$ to 50
2.   Start a Simulated Annealing process to minimise $f_S$
3.   Run SLMAP over the values obtained in 2.
4.   Compute approx. for $IDS^{n+1}$ and store in ListIDS
5. Compute MajIDS, the Majority Vector of all members of ListIDS
6. Get candidate values for $IDS^{n+1}$, $IDS_0$ and $IDS_1$
7. Compute correlation of these values with MajIDS, $\rho_0$ and $\rho_1$
8. If $\rho_0 > \rho_1$ then $IDS^{n+1}$=$IDS_0$ else $IDS^{n+1}$=$IDS_1$
```

Fig. 2. Outline of the traceability attack.

V. Concluding Remarks

In this paper we have presented a very efficient and effective black-box attack against a novel and quite interesting ultra-lightweight authentication protocol called SLMAP.

This attack is implemented by using a non-standard cryptanalytic technique based on the use of a Simulated Annealing algorithm, which is able of setting up a traceability attack with a quite high success probability (around 95%). We, however, make no optimality claim whatsoever, and believe that even better traceability attacks could be mounted by refining the presented black-box technique, either by a slight change in the SA parameters, or possibly even by an alternative analytic formulation.

Studying other lightweight protocols like SASI [6] with similar techniques is a future and interesting research direction. Another promising research line is to mount similar attacks not only against the traceability property of a security protocol, but against more classical security objectives such as their secrecy or authentication capabilities, possibly by trying to recover the value of the secret identifier of the tag (ID), which is the value all the protocols are designed to conceal. This will be considered in future works.

References

[1] Lars R. Knudsen and Willi Meier. "Cryptanalysis of an Identification Scheme Based on the Permuted Perceptron Problem" In *Advances in Cryptology - EUROCRYPT 1999*, LNCS 1592, pp. 363–374. Springer-Verlag, 1999.

[2] John A. Clark and Jeremy L Jacob. "Fault Injection and a Timing Channel on an Analysis Technique". In *Advances in Cryptology - EUROCRYPT 2002*, LNCS 2332, pp. 181–196. Springer-Verlag, 2002.

[3] John A. Clark and Jeremy L Jacob. "Protocols are Programs Too: the Meta-heuristic Search for Security Protocols". Special Issue on Metheuristics for Software Engineering. *Information Software Technology* 43(14):891-904, December (2001).

[4] Hao Chen, John A. Clark, Jeremy Jacob. "Human competitive security protocols synthesis". *Proceedings of the 8th annual conference on Genetic and evolutionary computation (GECCO 2006)*, pp. 1855–1856.

[5] Julio Cesar Hernandez-Castro, Juan M. Estevez-Tapiador, Arturo Ribagorda: "Non-standard Attacks against Cryptographic Protocols, with an Example over a Simplified Mutual Authentication Protocol". *MCO 2008*, pp. 589–596.

[6] Hung-Yu Chien. "SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity". *IEEE Transactions on Dependable and Secure Computing* 4(4):337–340, Oct.-Dec. 2007.

[7] David Pointcheval. "A New Identification Scheme Based on the Perceptron Problems." In *Advances in Cryptology - EUROCRYPT 1995*. LNCS 2199. Springer-Verlag, 1995

[8] Tieyan Li, Guilin Wang. "SLMAP - A Secure ultra-Lightweight RFID Mutual Authentication Protocol". In *Proceedings of Chinacrypt'07*, Oct. 19-22, 2007. Cheng Du, China.

[9] A. Juels and S.A. Weis. "Defining Strong Privacy for RFID". *Proceedings of IEEE PerCom'07*, pp. 342–347, 2007. Full version available at IACR ePrint Archive, http://eprint.iacr.org/2006/137, 7 April 2006.

[10] Raphael C. Phan. "Cryptanalysis of a New Ultralightweight RFID Authentication Protocol - SASI". *IEEE Transactions on Dependable and Secure Computing*. 2008.

[11] Ton van Deursen and Sasa Radomirovic. "On a New Formal Proof Model for RFID Location Privacy". Cryptology ePrint Archive: Report 2008/477. Available online at http://eprint.iacr.org/2008/477.

[12] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M. Robshaw, and Y. Seurin. "Hash Functions and RFID Tags: Mind the Gap". E. Oswald and P. Rohatgi (Eds.): *CHES 2008*, LNCS 5154, pp. 283–299. Springer-Verlag, 2008.

[13] Makoto Matsumoto, Takuji Nishimura. "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator". *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 8(1):3–30, Jan. 1998.

[14] S. Karthikeyan and M. Nesterenko. "RFID Security without Extensive Cryptography". In Proc. of SASN'05, 2005.

[15] D. Nguyen Duc, J. Park, H. Lee, and Kwangjo K. "Enhancing security of EPCglobal Gen-2 RFID tag against traceability and cloning". In Proc. of Symposium on Cryptography and Information Security, 2006.

[16] Y. Cui, K. Kobara, K. Matsuura, and H. Imai. "Lightweight asymmetric privacy-preserving Authentication protocols secure against active attacks". In Proc. of PerSec'07. IEEE Computer Society, 2007.